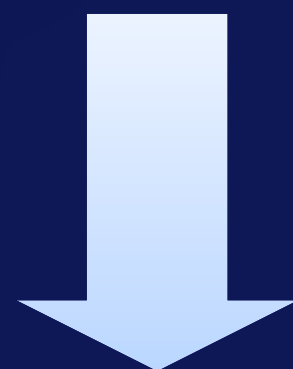


云原生时代应用架构的演进

Application Architecture Evolution in Cloud Native Era

许晓斌
阿里云高级技术专家

云原生已经进入了 **PaaS上云** 为主的阶段



IaaS 上云

PaaS 上云

计算、存储、网络
对应用无侵入

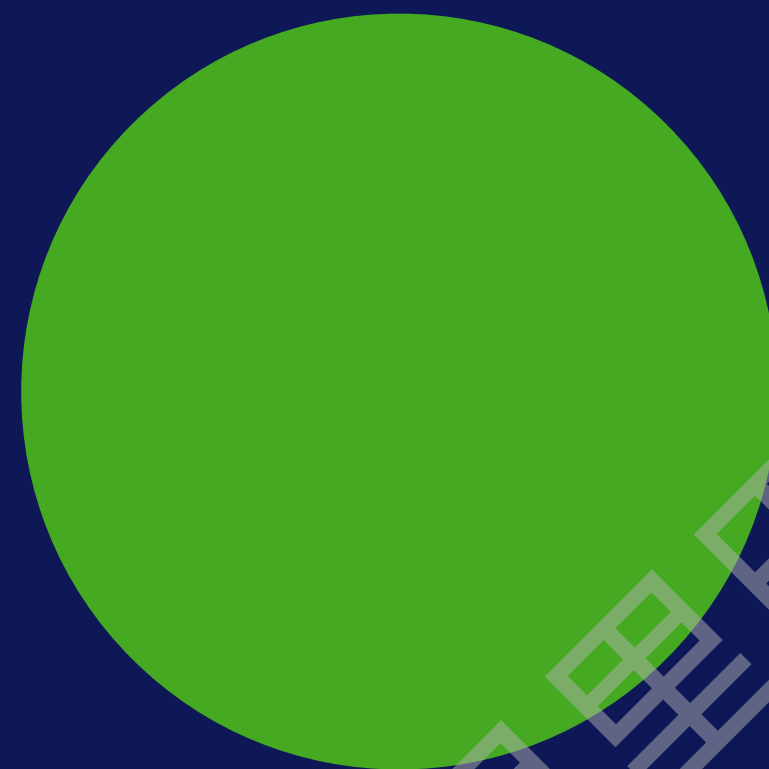
数据库、中间件、监控、应用生命周期托管
直接影响应用架构

云原生技术

	微服务	12 Factor Apps	容器	BaaS (API)	GitOps, IaC	Service Mesh
应用架构						
生命周期管理						
流量管理						
基础设施及依赖						

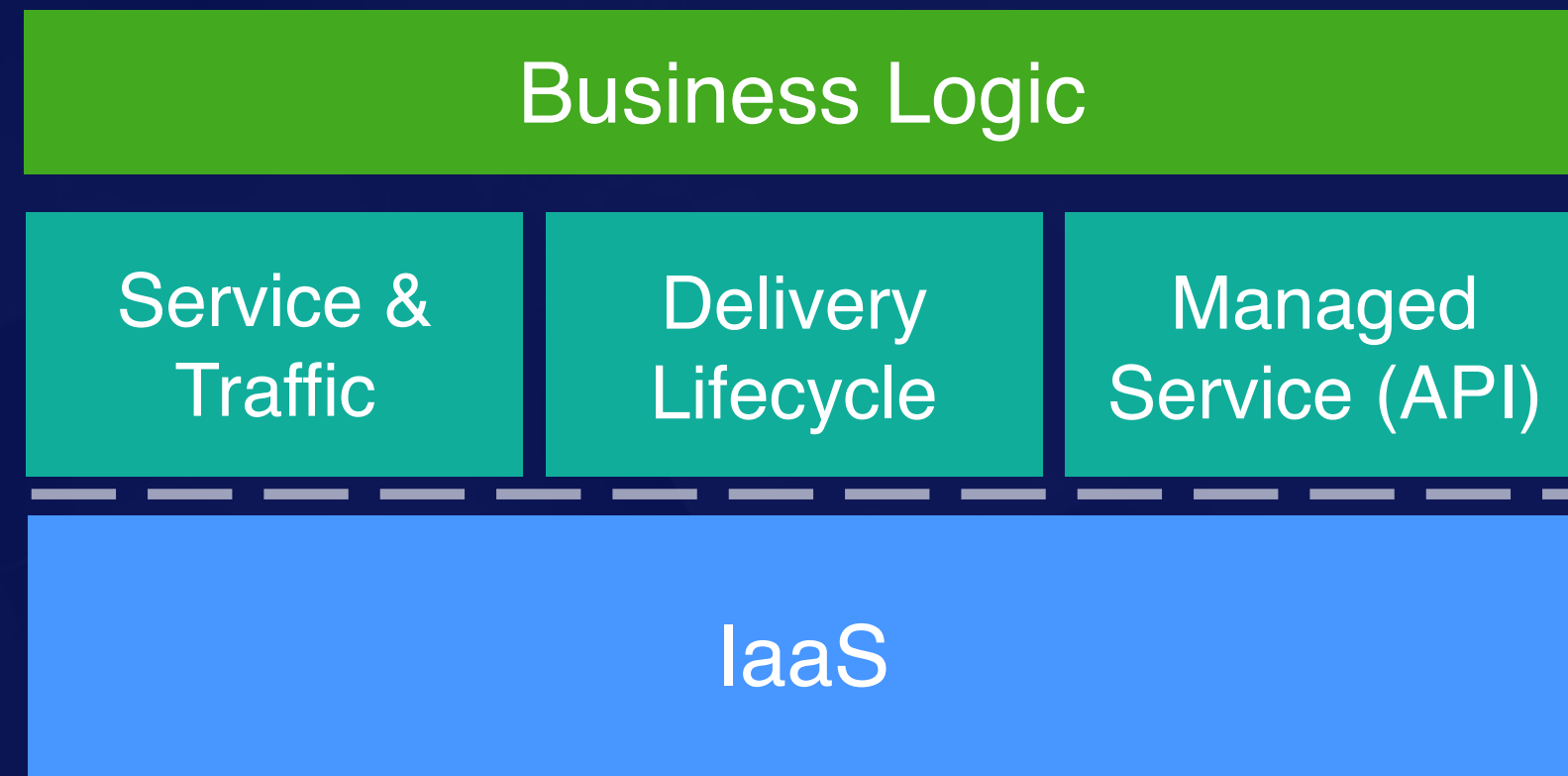
业务关心的是交付速度

1. 标准化平台/服务和应用的协议
2. 与业务无关能力进一步解耦至平台
3. 应用架构升级以适应 1, 2



业务交付速度

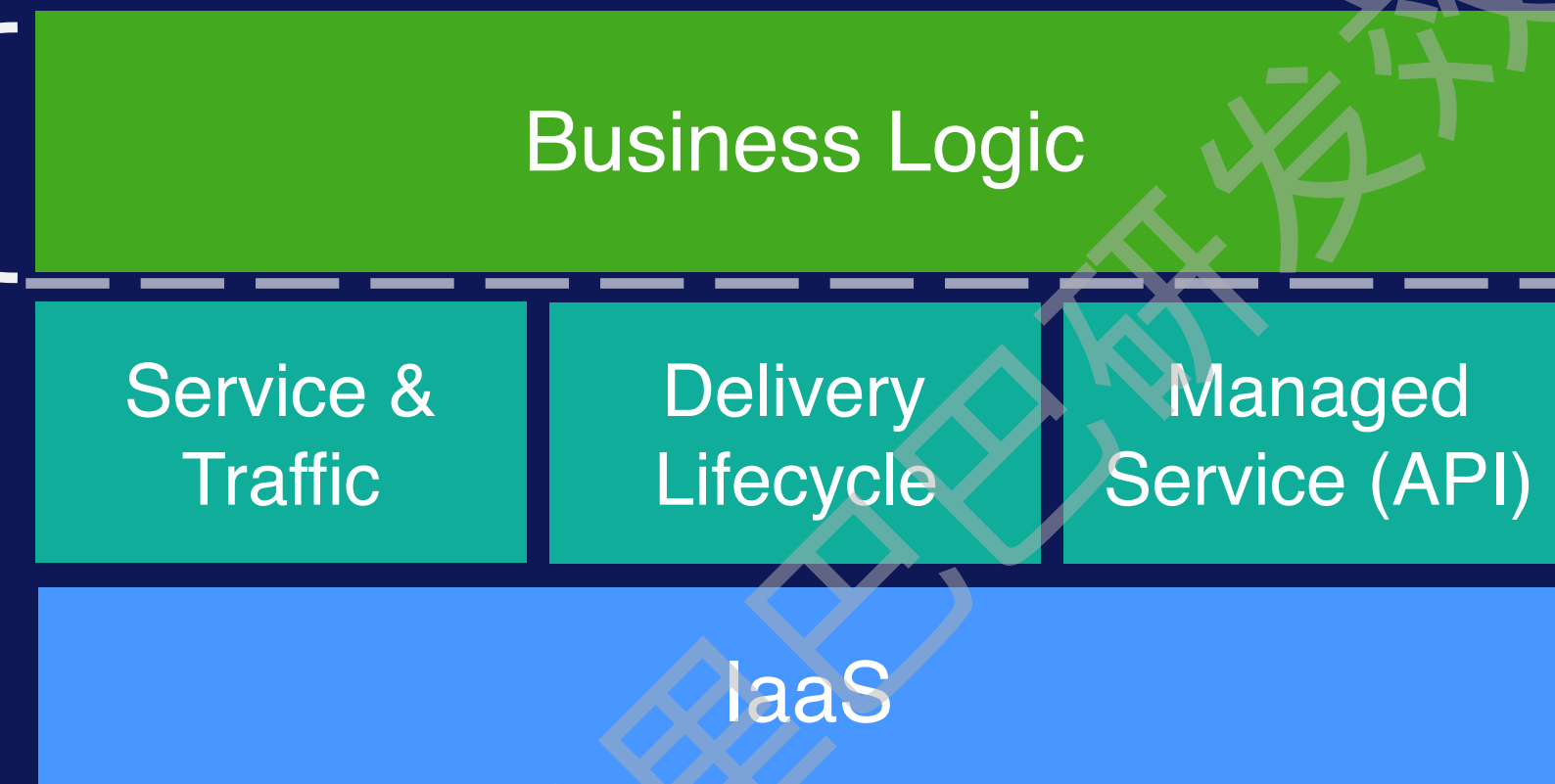
IaaS 上云阶段



研发需要关心

研发需要关心

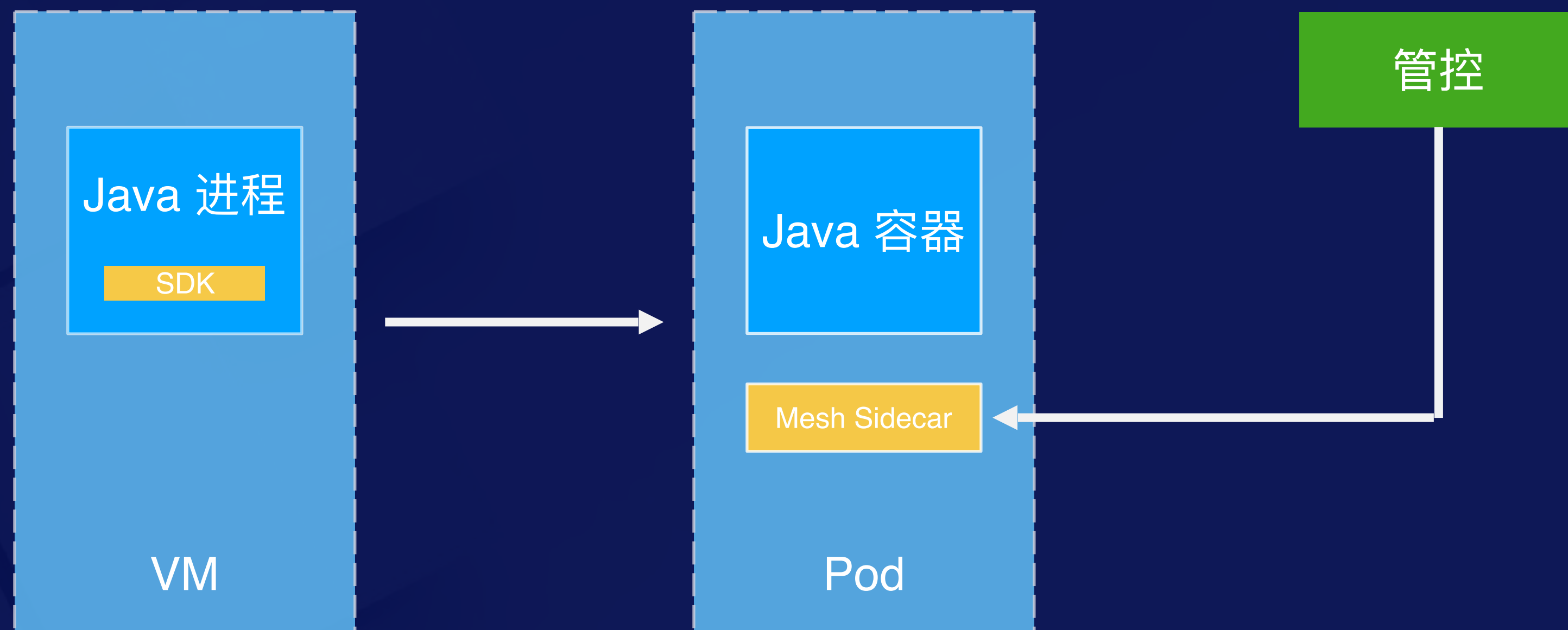
云原生上云阶段



核心问题

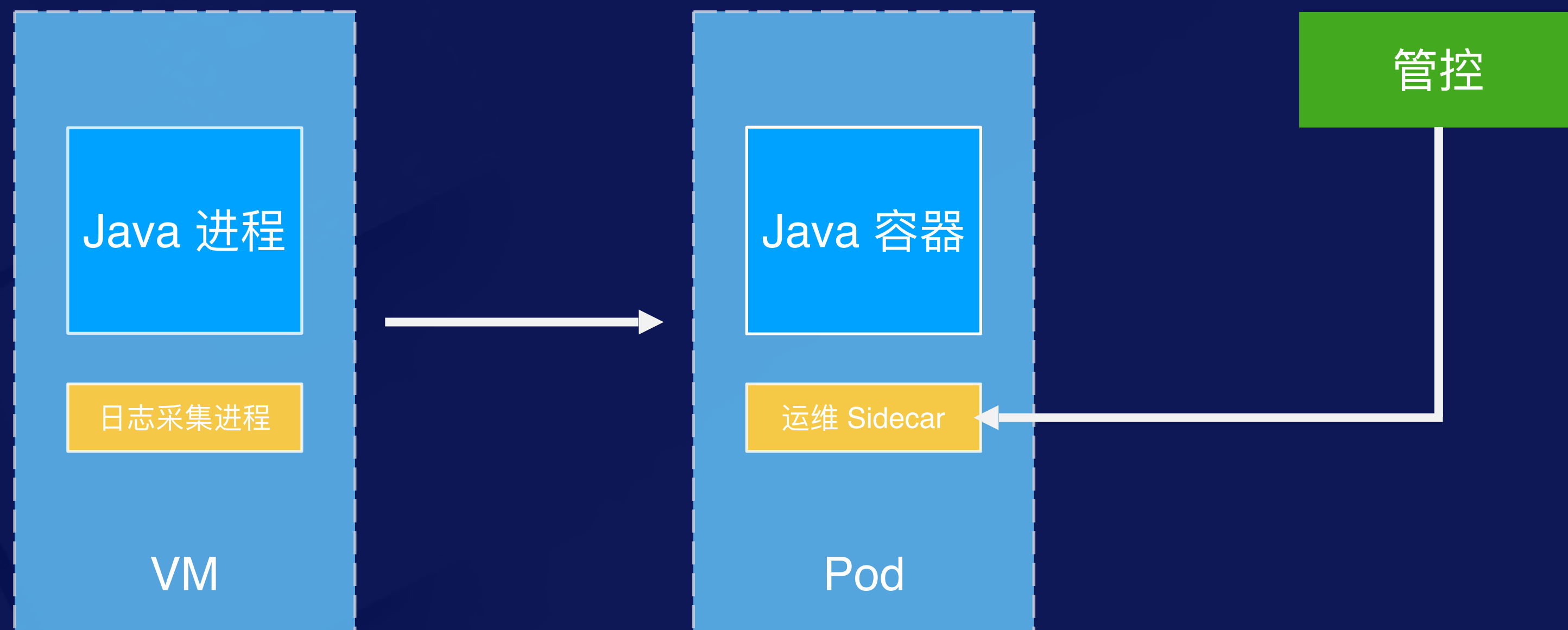
1. 业务无关能力如何解耦至平台？
2. 平台和业务（应用）之间的协议如何定义？
3. 应用架构需要如何适应？

Example: Service Mesh 把服务发现和流量从业务剥离



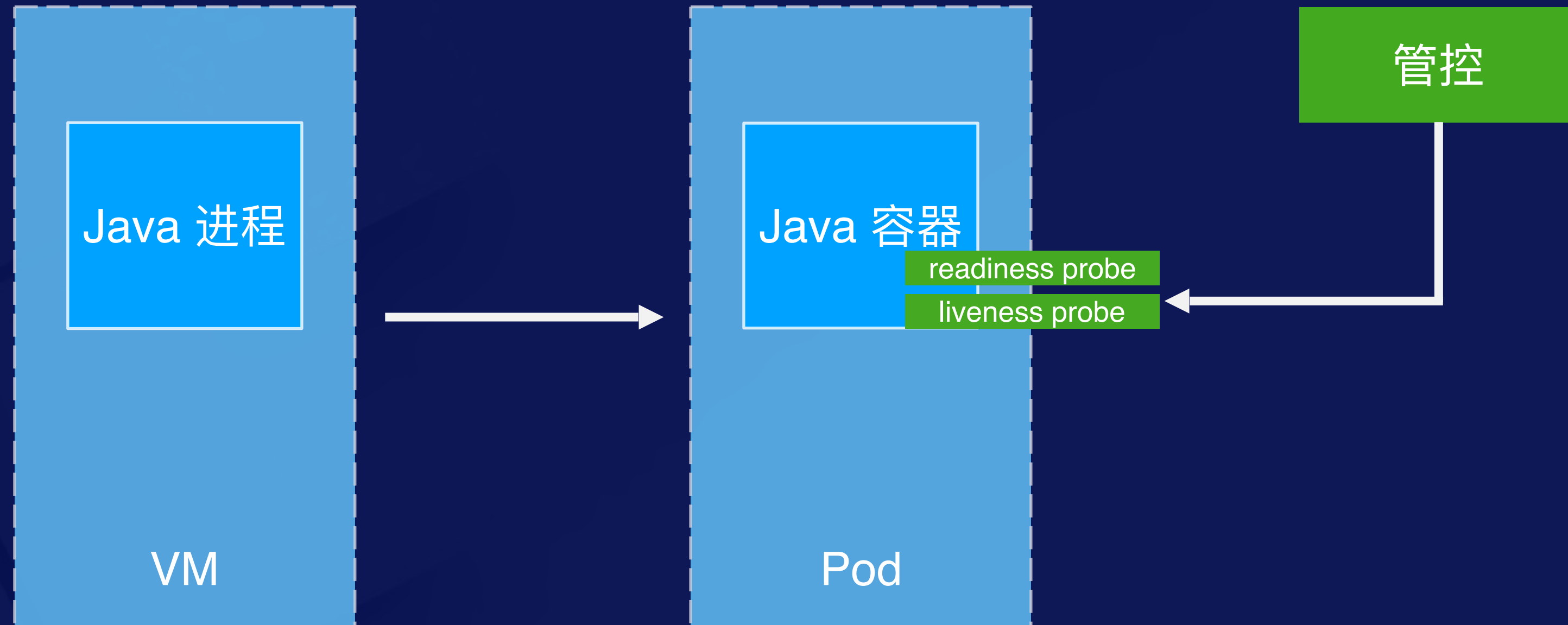
1. 服务注册 & 发现
2. 流量路由
3. 流量回放
4. 发布过程中流量控制

Example: 轻量化容器把日志采集从业务中剥离



1. 资源隔离
2. 独立升级

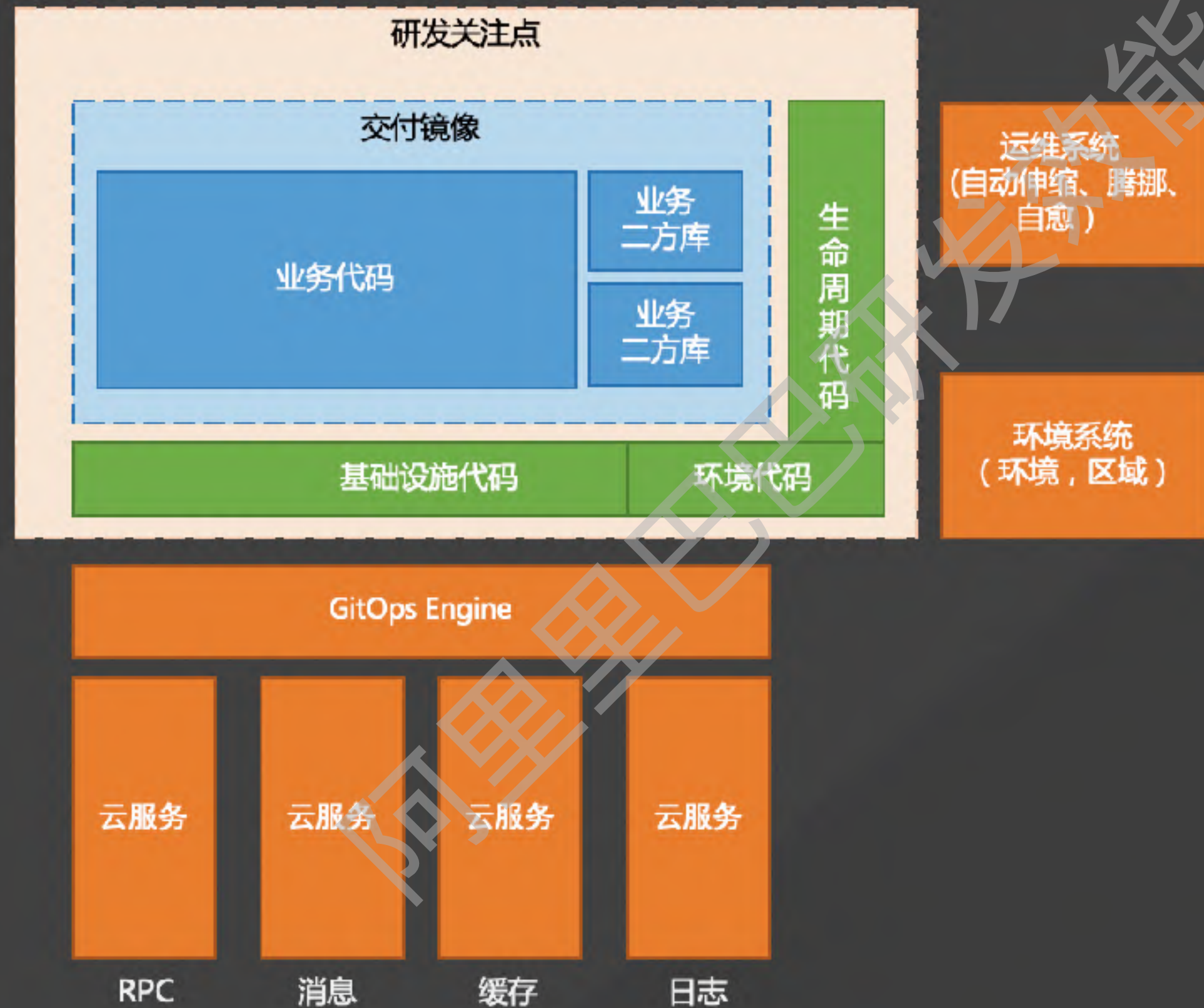
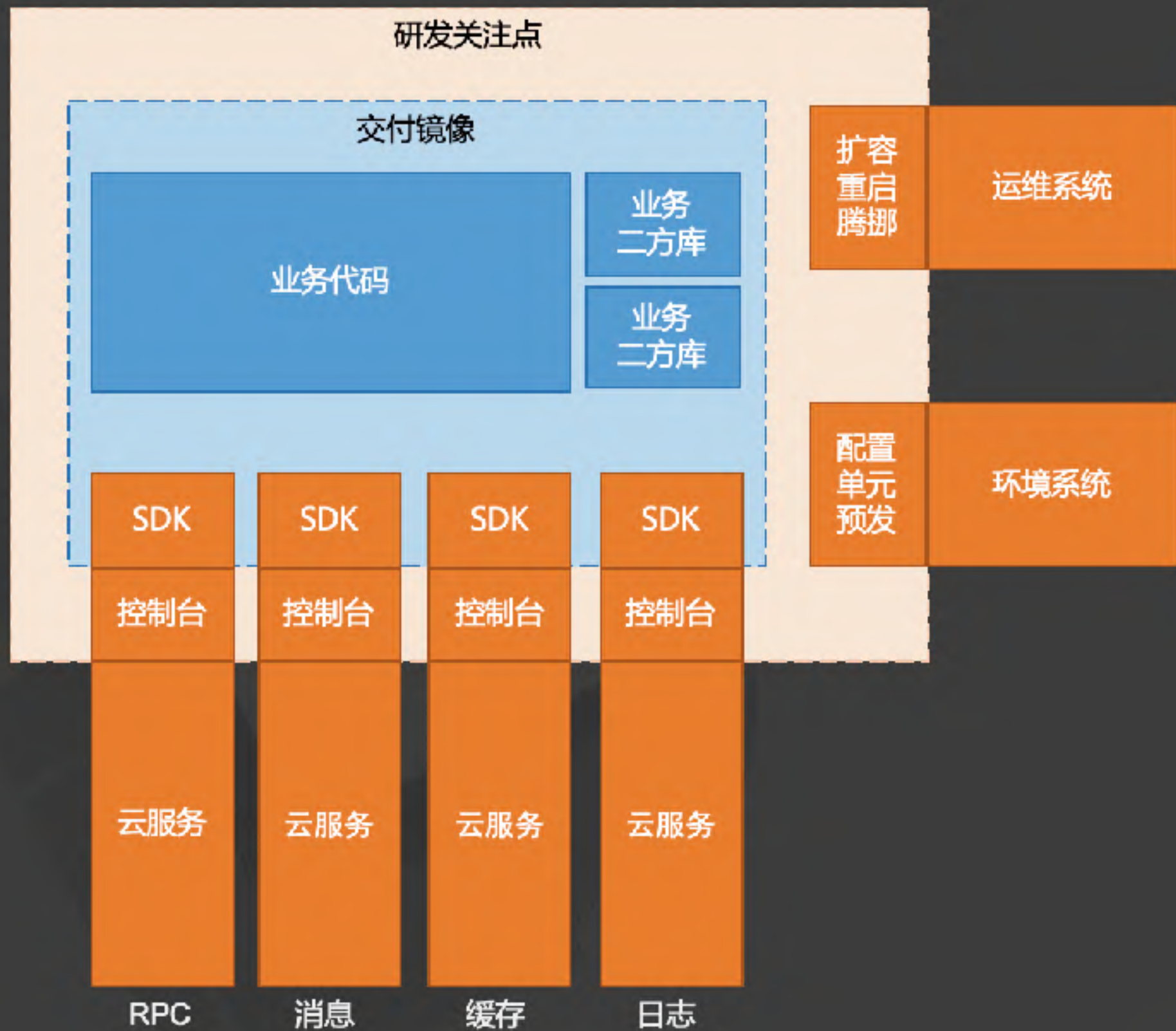
Example: 业务提供探针, 让平台实现生命周期管理



1. 自动弹性
2. 自动腾挪
3. 自动重启 (自愈)

协议 (Contract) = API + Configuration

1. Docker (& OCI) 是标准的软件交付 API
2. 作为 RPC 协议, 开源的 GRPC/DUBBO 优于私有的 HSF
3. 作为缓存协议, 开源的 Redis 优于私有的 Tair
4. 微软的 Dapr 尝试基于 sidecar 架构将 API 标准化到 HTTP/GRPC 层, 以去 SDK, 并支持多语言
5. Terraform, Pulumi 等 IaC 产品, 通过配置语言声明基础设施
6. GitOps 进一步的使用代码声明环境、发布流程、发布策略内容



今天我们在以类似**汇编**的形式使用云

云原生就是定义业务和云之间的**契约**
并围绕这个契约构建**高级编程语言和工具**

阿里巴巴研发效能峰会

Thank You



阿里巴巴研发效能峰会